UCRL-TR-205652

Certification of Completion of Level-2 Milestone 405: *Deploy Next-Generation Data Management and Analysis Environment*

Jeffery W. Long, Randall J. Frank, Eric Brugger, Steven Louis

July 30, 2004

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48

Introduction

This summary report describes data management and visualization activities in the Advanced Simulation and Computing (ASC) program at Lawrence Livermore National Laboratory (LLNL). The report covers the period from approximately October 2003 to June 2004 and describes activities within the Visual Interactive Environment for Weapons Simulation (VIEWS) ASC program element. This report and the references herein are intended to document the completion of the following Level 2 Milestone from the ASC FY04–05 Implementation Plan, due at the end of Quarter 3 in FY04:

> **Milestone**: 405
>
> **Title**: Deploy Next-Generation Data Management and Analysis Environment
>
> **Category**: Campaign 11—NA114, Advanced Simulation and Computing
>
> **ASC Program Element**: Simulation and Computer Science/VIEWS

The ASC program conducts predictive simulation in support of the nation's Stockpile Stewardship Program. VIEWS focuses on the problem of "seeing and understanding" the results of multi-teraOPS simulations and comparing results across simulations and between simulations and experiments.

This milestone covers the deployment of software packages that together form an end-to-end simulation post-processing environment for weapon scientists. These software packages represent formal development efforts and products within VIEWS, and are targeted at specific user needs. In addition to providing core capabilities, some of these products also serve as vehicles for delivering the results of VIEWS research to end-users.

This documentation is organized as a series of short development overviews that include pointers to publications, Web pages, and points of contact where additional details can be found. The report is not intended to be comprehensive, but it does represent the major data management and analysis activities in the VIEWS program element specific to this Level 2 Milestone.

TeraScale Browser

Description

The TeraScale Browser (TSB) is an end-user tool that provides interactive navigation through time and space of multi-terabyte datasets with adaptive resolution control. It is the result of the research efforts within VIEWS for data access, volume rendering and surface display.

In a visualization system embracing view-dependent, multi-resolution visualization techniques, the effective management of data staging, pipelining, bandwidths and cache systems is critical in maintaining interactivity. The TeraScale Browser begins with the assumption that the data can never fit into memory and that the Browser's major task is to dynamically balance the allocation of both local and remote system resources to provide the highest fidelity display possible, given system resource constraints and user demands. The goal is to maintain scalable performance throughout the entire system, not just a selected fraction of the system, to ensure that interactive user expectations may be met. To achieve this goal, we sacrifice much of the broad, detailed functionality present in classical visualization tools for interactive scalability, focusing on techniques for which algorithms with controlled resource footprints exist. Scalable algorithms for slicing at arbitrary orientations, families of surfaces (iso-contour and material boundaries) and volume rendering have been integrated into the tool. Many of these algorithms require substantial preprocessing to compute specialized data structures for interactive display. To accommodate this requirement for such large one-time computations, a three-phase approach to large data visualization is used: prepare, browse, and select.

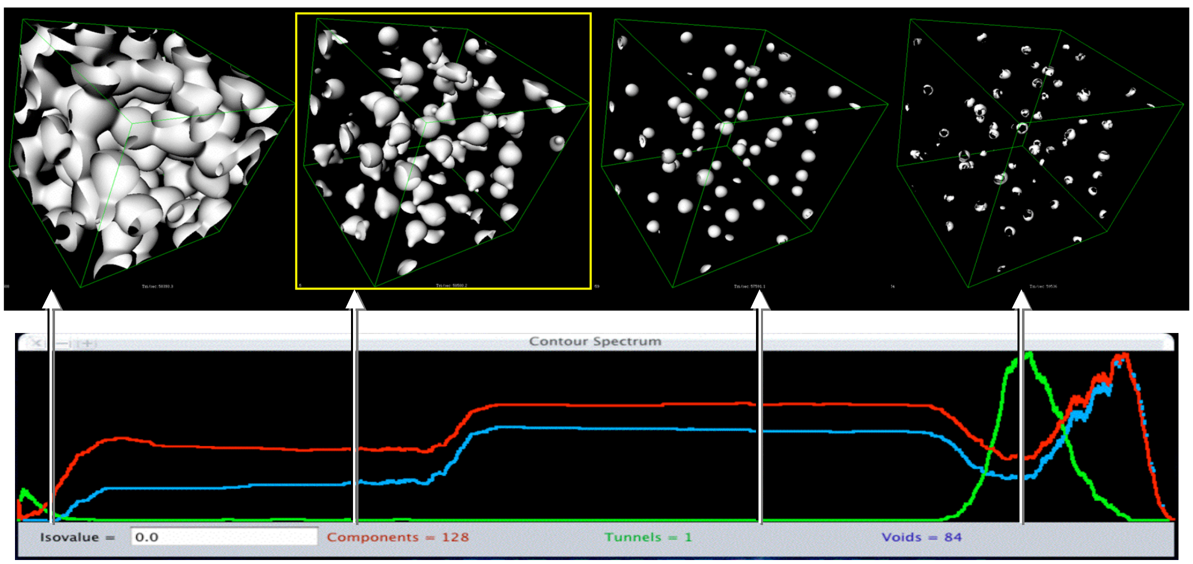
Milestone Accomplishments

The following features were added to the TSB to complete its portion of the end-to-end simulation post-processing environment:

- Incorporated first-generation topological analysis capability, which provides a high-level view of temporal evolution and a mechanism for finding "interesting" phenomena. It can also be used for data comparison and normalization. Further, the ability to compute and display temporal contour spectra on an unstructured mesh was demonstrated.
- Built distributed data preparation infrastructure, which enables the "prepare" step to be performed in parallel on remote hosts best suited for this work. A local GUI (graphical user interface) can launch the remote data preparation processes, thus simplifying the procedure for the user.
- Revamped graphical user interface to be more consistent and friendly. A C++ and Motif approach was used to consolidate the interface between different applications, assuring the user of a seamless experience with the various program elements.
- Released new version, 1.3.4, which features performance enhancements to exploit graphics processing units. The utilization of inexpensive GPUs (graphics processing

units) on local workstations presents tremendous possibilities for improved rendering and manipulation capabilities.

Contact: Dave Bremer, dbremer@llnl.gov
Web Page: http://www.llnl.gov/icc/sdd/img/terascale_views.shtml



Example of temporal summary of raw data, using contour spectra

VisIt

Description

VisIt is an interactive parallel visualization and graphical analysis tool for viewing scientific data on Unix and PC platforms. Users can quickly generate visualizations from their data, animate them through time, manipulate them, and save the resulting images for presentations. VisIt contains a rich set of visualization features so that you can view your data in a variety of ways. It can be used to visualize scalar and vector fields defined on two- and three-dimensional (2D and 3D) structured and unstructured meshes. VisIt was designed to handle very large data set sizes in the terascale range and yet can also handle small data sets in the kilobyte range.

VisIt's visualization capabilities are primarily grouped into two categories: plots and operators. Plots are used to visualize data and include boundary, contour, curve, mesh, pseudo-color, streamline, surface, vector, and volume. Operators consist of operations that can be performed on the data prior to visualization. Some examples include slice, index select, iso-surface, onion peel, reflect, threshold, and part selection.

VisIt is also a powerful analysis tool. It provides support for derived fields, which allow new fields to be calculated using existing fields. For example, if a dataset contains a velocity field, it is possible to define a new field that is the velocity magnitude. VisIt's quantitative analysis tools include: line-out, which allows you to create curves from higher dimensional datasets by interactively defining lines using the mouse, and pick and query, which allows you to query the dataset by clicking on images. VisIt also supports a generalized query interface, which allows you to query derived quantities such as volume or surface area.

VisIt employs a distributed and parallel architecture in order to handle extremely large data sets interactively. VisIt's rendering and data processing capabilities are split into viewer and engine components that may be distributed across multiple machines.

Milestone Accomplishments

VisIt released numerous new capabilities during the period October 1, 2003 to July 1, 2004 over 9 releases culminating in the VisIt 1.3.2 release on June 28, 2004. Highlights include:

- Implemented a new material interface reconstruction algorithm that reduced the execution time, memory usage, and rendering time.
- Enhanced the volume plot to use 2d and 3d texture mapping capabilities of new graphics cards to improve the accuracy and rendering speed.
- Completed a native Mac OS X port.
- Generalized the handling of data from multiple databases. This allows multiple, unrelated, time-varying databases to coexist in the same visualization window. It also

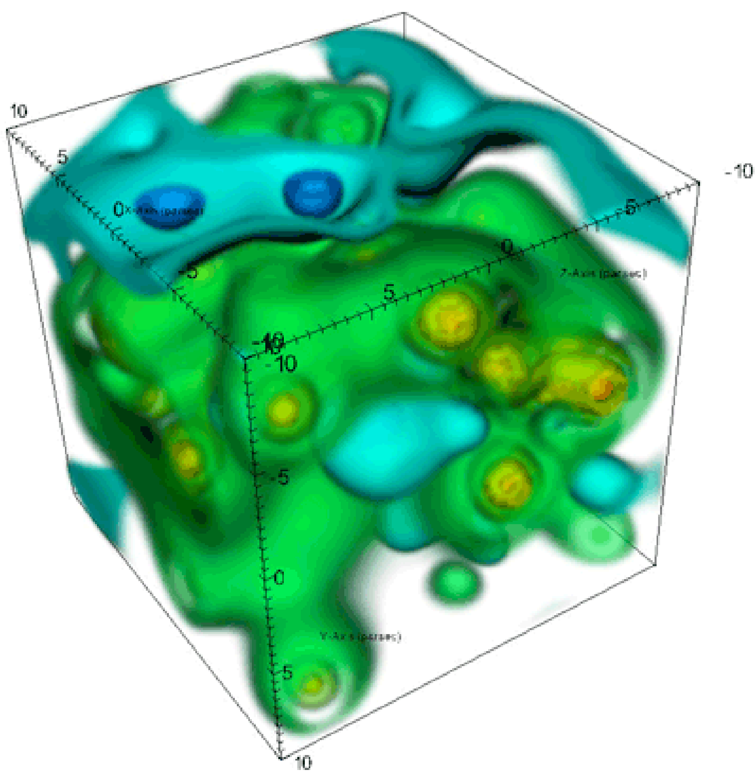
enables the creation of time or cycle correlations between databases, which allow the user to lock the plots from multiple databases together in time.

- Developed framework for generalized annotation capabilities, allowing users to more easily attach labels and comments to graphics.
- Significantly improved the rendering performance. This included tuning the rendering code and improving the performance of display list generation. This resulted in performance gains of 2x to 75x depending on the data and hardware.

There were also a number of accomplishments having to do with the deployment of VisIt to end users.

- Created a one-day class on the use of VisIt, and due to popular demand presented it multiple times.
- Greatly expanded the nightly regression suite to improve the quality of VisIt.

Contact: Eric Brugger, brugger1@llnl.gov
Web Page: http://www.llnl.gov/visit/



VisIt image illustrating the result of setting the volume transform function used to map field values to opacity and color.

SimTracker

Description

SimTracker is a web-based application for managing and organizing simulation results and other data. It helps users archive, sort, copy, share, search, and monitor their collections of data.

SimTracker collects and presents information about calculations at a number of levels. At the top-most level, the table of contents page shows a single record for each simulation run. One can see specific details of a simulation by going to the results page, which contains key metadata, links to input and output files, and a series of images generated from the simulation. Drilling down further, one sees cycle pages containing information about individual cycles in the simulation.

SimTracker's web interface enables users to perform a wide variety of operations on their simulation data quickly and conveniently. In addition to organizing, browsing, and searching simulation results, SimTracker allows users to easily and safely share results with colleagues, to monitor running simulations, and to manage local and remote data.

SimTracker allows users to annotate simulations in a number of ways, such as by adding comments associated with a particular simulation, image, cycle, or file. Users can associate viewgraphs, images, or any other type of data with their SimTracker summaries. Using SimTracker's web interface, users can easily archive results in HPSS (High Performance Storage System), or move files to another server for doing additional post-processing. SimTracker offers a variety of useful search capabilities that let users find summaries of interest. For instance, one can find all summaries that contain a particular keyword, or which match an arithmetic expression.

Milestone Accomplishments

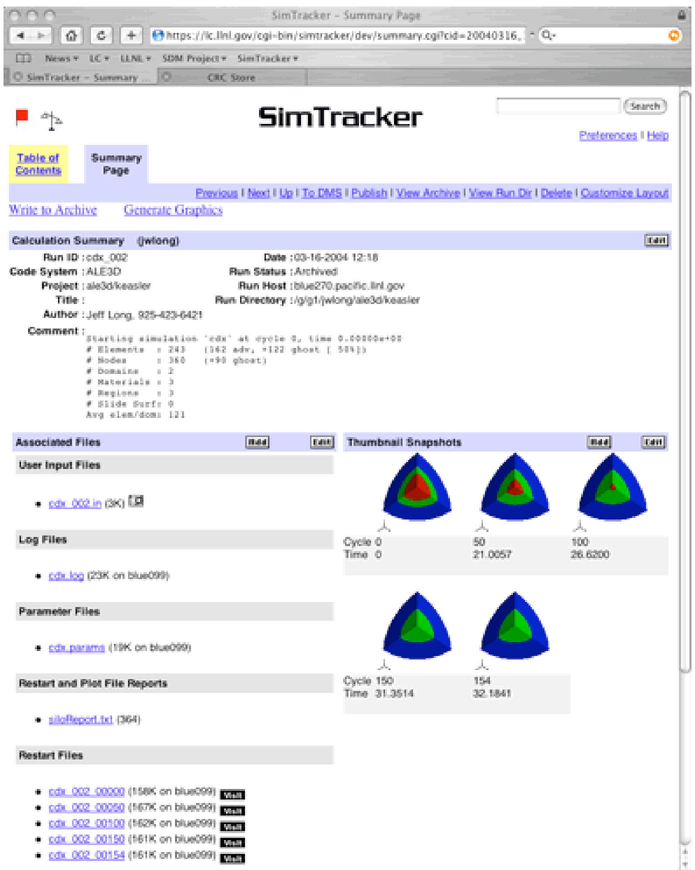
The following features were added to SimTracker to complete its portion of the end-to-end simulation post-processing environment:

- Integrated with visualization and rendering capabilities on a commodity cluster architecture. SimTracker users now can launch VisIt's front-end on their local desktop, and automatically launch VisIt's back-end on the host where the data resides. This simplifies the process of analyzing data, and provides a seamless transition from browsing metadata to interpreting data.
- Demonstrated new comparison framework enabling users to compare simulation data. This new framework supports pluggable comparison operators that range from simple to complex. Simple operators include textual "diffs", which allow comparisons of text files such as input decks from two simulations, and image diffs, which allow thumbnail and other images from multiple simulations to be viewed side by side or as subtracted images. More complex operators support differencing of variables within binary data files. The framework also provides a means for later incorporating "smart

comparisons," e.g., by allowing users to compare characterizations of their data generated by feature detection and extraction algorithms.

- Implemented persistent task management capability. This provides two key functions: persistent data operations and a detailed record of operations and their status. The persistence is important to ensure that requested transfers or deletions are completed successfully, even if networks or computing resources go offline during the operation. A complete and accurate picture of the status of each of these tasks is also provided, allowing users to monitor progress of active tasks and to review completed tasks.

Contact: Jeff Long, jwlong@llnl.gov
Web Page: http://www.llnl.gov/icc/sdd/img/scientific.shtml



SimTracker results page, illustrating metadata automatically gathered.

Hopper

Description

Hopper is an interactive tool that allows users to transfer and manipulate files and directories by means of a graphical user interface. Users can connect to and manage resources using the major file transfer protocols, including SSH, FTP, and SFTP. Hopper also supports the high-performance HTAR (LLNL's HPSS TAR utility), NFT (LLNL's persistent file transfer utility, and HPSS protocols available to ASC scientists at LLNL.

Hopper is implemented in Java and can therefore be run on everything from desktop machines (Windows, Unix, or OS X) to large production machines. This means that regardless of platform and transfer protocol, users can use the familiar and consistent graphical Hopper interface to manage and transfer their files.

Hopper uses abstraction layers to extend its capabilities. The most important abstraction is the virtual file system, which provides a uniform interface to all of the various underlying technologies supported by Hopper. Thus far ten virtual file system interfaces have been developed, including FTP, SSH, SFTP, and TAR.

Hopper's copy subsystem transfers data between virtual file systems, providing recursion, two-hop or three-hop transfers, etc. as needed. The utility of this architecture is exhibited when transferring items between dissimilar file systems. For instance, one can transfer files from inside a tar file on one remote host to a directory on another remote host connected via ftp.

Hopper is highly concurrent, supporting any number of simultaneous transfer or search operations, all without interfering with the user's ability to interact with his or her directory windows. A set of listeners and thread managers keep track of the various threads and make sure information is properly collected and reported.

To optimize performance, operations such as searches are implemented by running native Unix utilities like "find" or "locate". In situations where these utilities are not available, internal Java code is used to perform the same functions. This allows Hopper to perform searches on virtual file systems, such as FTP and TAR, which do not have intrinsic search capabilities.

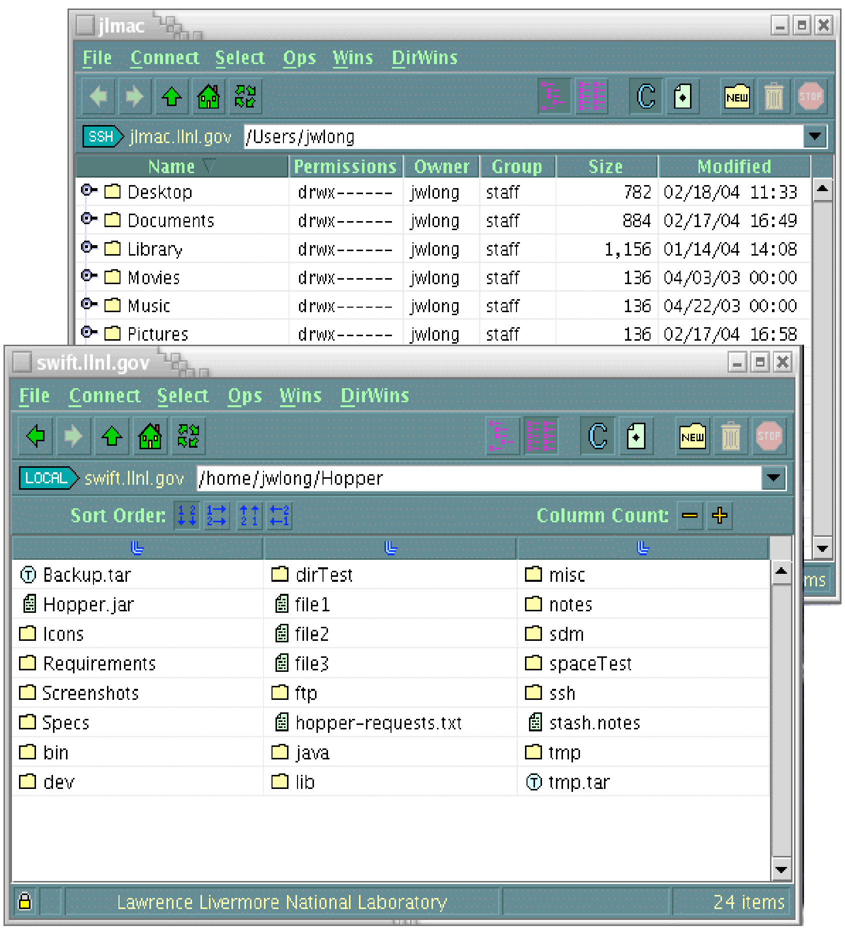
Milestone Accomplishments

The following features were added to Hopper to complete its portion of the end-to-end simulation post-processing environment:

- Released two production versions to users, accompanied by presentations, technical bulletins, and a full-featured web site.
- Extended remote access support to include Secure FTP and Endeavor (NFT). The former provides secure, encrypted transfers while the latter provides parallel, persistent transfers.

- Developed popular access checking capability, allowing a user to easily identify others who have access to a selected file or directory.
- Incorporated sophisticated authentication management logic, moving towards the goal of a single sign-on system.
- Integrated with metadata infrastructure, allowing users to create and browse annotations for files and directories.
- Significantly enhanced and optimized graphical user interface. Easy access features were added to benefit vision-impaired users, GUI performance was improved by a factor of five, and a powerful selection capability was added.
- Extended support for HTAR and TAR files, including creating HTAR and TAR files on remote hosts, and opening and creating compressed TAR files.

Contact: Jeff Long, jwlong@llnl.gov
Web Page: http://www.llnl.gov/hopper/



Hopper's tabular and detailed directory views.

Appendix A. Additional Reference Materials

TSB & Contour Spectrum:
- http://www.llnl.gov/icc/sdd/img/viz.shtml
- http://www.llnl.gov/icc/sdd/img/terascale_views.shtml
- http://www.llnl.gov/icc/sdd/img/terascale.shtml
- TSB man page
- Slides presented at ASC PI meeting
- http://pascucci.org/pdf-papers/spectrum.pdf (background)

VisIt:
- http://www.llnl.gov/visit/
- Copy of "About VisIt" page:
  - http://www.llnl.gov/visit/about.html
- Getting Started manual:
  - ftp://ftp.llnl.gov/pub/visit/visit1.1/GettingStarted.pdf

SimTracker:
- https://lc.llnl.gov/simtracker/
- User's Manual
- Viewgraphs

Hopper:
- http://www.llnl.gov/hopper/
- Viewgraphs from LC customer meeting:
  - http://www-r.llnl.gov/icc/viewgraphs/viewgraphs04/apr/smith/smith0404.pdf
- README file: http://www.llnl.gov/hopper/hopper_readme.html